



Coarse-grained localized diffusion

Guy Wolf^a, Aviv Rotbart^a, Gil David^b, Amir Averbuch^{a,*}

^a School of Computer Science, Tel Aviv University, Tel Aviv 69978, Israel

^b Department of Mathematics, Program in Applied Mathematics, Yale University, New Haven, CT 06510, USA

ARTICLE INFO

Article history:

Received 23 August 2011

Revised 23 February 2012

Accepted 26 February 2012

Available online 2 March 2012

Communicated by Charles K. Chui

Keywords:

Diffusion maps

Localized diffusion folders

Coarse-graining

Dimensionality reduction

ABSTRACT

Data-analysis methods nowadays are expected to deal with increasingly large amounts of data. Such massive datasets often contain many redundancies. One effect from these redundancies is the high dimensionality of datasets, which is handled by dimensionality reduction techniques. Another effect is the duplicity of very similar observations (or data-points) that can be analyzed together as a cluster. We propose an approach for dealing with both effects by coarse-graining the popular *Diffusion Maps* (DM) dimensionality reduction framework from the data-point level to the cluster level. This way, the size of the analyzed dataset is decreased by only referring to clusters instead of individual data-points. Then, the dimensionality of the dataset can be decreased by the DM embedding. We show that the essential properties (e.g., ergodicity) of the underlying diffusion process of DM are preserved by the coarse-graining. The affinity that is generated by the coarse-grained process, which we call *Localized Diffusion Process* (LDP), is strongly related to the recently introduced *Localized Diffusion Folders* (LDF) [G. David, A. Averbuch, Hierarchical data organization, clustering and denoising via localized diffusion folders, Appl. Comput. Harmon. Anal. (2011), in press] hierarchical clustering algorithm. We show that the LDP coarse-graining is in fact equivalent to the affinity-pruning that is achieved at each folder-level in the LDF hierarchy.

© 2012 Elsevier Inc. All rights reserved.

1. Introduction

Massive high-dimensional datasets have become an increasingly common input for data-analysis tasks. When dealing with such datasets, one requires a method that reduces the complexity of the data while preserving the essential information for the analysis. One approach for obtaining this goal is to analyze sets of closely-related data-points, instead of directly analyzing the raw data-points. A recent approach for obtaining such an analysis is the Localized Diffusion Folders (LDF) method [1]. This method recursively prunes closely-related clusters, while preserving the information about local relations between the pruned clusters.

The Diffusion Maps (DM) framework [2,3] provides an essential foundation for LDF to succeed. This framework is based on defining similarities between data-points by using an ergodic Markovian diffusion process on the dataset. The ergodicity of this process ensures it has a stationary distribution and numerically-stable spectral properties. The transition probability matrix of this process can be used to define diffusion affinities between data-points. The first few eigenvectors of this diffusion affinity kernel represent the long-term behavior of the process and they can be used to obtain a low-dimensional representation of the dataset, in which the Euclidean distances between data-points correspond to diffusion distances between their original (high-dimensional) counterparts. We present a coarse-graining of this diffusion process,

* Corresponding author. Fax: +972 3 6422020.

E-mail address: amir@math.tau.ac.il (A. Averbuch).

while preserving its essential properties (e.g., ergodicity). We show that this coarse-graining is equivalent to the pruning method that appeared in the LDF.

The LDF method performs an iterative process that obtains a folder hierarchy that represents the points in the dataset. Each level in the hierarchy is constructed by pruning clusters of folders (or data-points) from the previous level. The iterative process has two main phases in each iteration:

1. **Clustering phase:** the “shake & bake” method is used to cluster the folders (or data-points) of the current level in the hierarchy by using a diffusion affinity matrix.
2. **Pruning phase:** the clusters of the current level are pruned and given as folders of the next level in the hierarchy. The diffusion affinity is also pruned to represent affinities between pruned clusters (i.e., folders of the next level in the hierarchy) instead of folders in the current hierarchical level.

In this paper, we focus on exploring the pruning that is performed in the second phase of this process, while considering the clustering of the data, which may be performed by “shake & bake” process [1] or by another clustering algorithm, as prior knowledge.

Essentially, LDF provides an hierarchical data clustering with additional affinity information for each level in the hierarchy. Other examples of hierarchical clustering methods can be found in [4,5]. However, these methods are not related to DM and to its underlying diffusion process. Since we are mainly concerned with the pruning phase of the algorithm, the clustering aspect of LDF and its relation with these methods is beyond the scope of this paper. A detailed survey of clustering algorithms and their relation to LDF is provided in [1, Section 2].

While there are many empirical justifications for the merits of LDF and its utilization in various fields (e.g., unsupervised learning and image processing), it lacked theoretical justifications. In this paper, we introduce a coarse-graining of the underlying diffusion process of DM. The resulting coarse-grained process, which we call Localized Diffusion Process (LDP), preserves essential properties of the original process, which enable its utilization for dimensionality reduction tasks. We relate this process, or rather the diffusion affinity generated by it, to the one achieved by the LDF pruning phase. This relation adds the needed complimentary foundations for the LDF framework by providing theoretical justifications for its already-obtained empirical support. Additionally, the presented relation shows that the applications presented in [1] in fact demonstrate the utilization of the LDP for data-analysis tasks and the results presented there provide empirical support of its benefits.

A similar coarse-graining approach was presented in [6]. The approach there is based on a graph representation of the diffusion random-walk process. The clustering of data-points was performed by graph partitioning. Then, transition probabilities between partitions were achieved by averaging transition probabilities between their vertices. The resulting random-walk process maintains most of the spectral properties of the original diffusion process and its eigendecomposition can be approximated by the original spectral decomposition. However, the approximation error strongly depends on the exact partitioning used. In addition, since all the random-walk paths are considered in the averaging process, there is a limited number of viable time-scales (in the diffusion process) that can be used by this process before it converges to the averaging of the stationary distribution.

The presented coarse-graining process in this paper copes with the rapid convergence toward the stationary distribution by only preserving localized paths between clusters while ignoring paths that are “global” from the cluster point-of-view. While it is desirable that the clusters will be sufficiently coherent to consist of a continuous partitioning of the dataset and its underlying manifold, the properties of the presented coarse-graining process are neither dependent on such assumptions nor on the exact clustering method used.

An alternative approach for local sets considerations of data-points is to analyze them as patches on the underlying manifold of the dataset [7–10]. The relations between patches are represented by non-scalar affinities that combine the information about both geodesic proximity of the patches and the alignment of their tangent spaces. This approach was used in [9] to modify DM to preserve the orientation of the manifold through the embedding process. A more comprehensive utilization of this approach was presented in [7] and [10], where affinities between patches were defined as matrices that transform vectors between tangent spaces. Parallel transport operators were used in [10], with the resulting affinity block matrix being related to the connection-Laplacian. Linear projections were used in [7] and further explored in [8], where the resulting diffusion process was shown to propagate tangent vectors on the manifold.

Both discussed methods in [7,10] lead to an embedded tensor space instead of a vector space. Also, the resulting diffusion process is not necessarily ergodic and may not have a stationary distribution. Therefore, they do not preserve one of the crucial properties of the diffusion process used in DM. The approach used in this paper produces a scalar-affinity matrix between closely-related clusters of data-points. It neither depends explicitly on the existence nor on the knowledge of the (usually unknown) underlying manifold of the dataset. The resulting diffusion process is similar to the one used in DM (for data-points), and it preserves the essential properties of that diffusion process. Finally, the same spectral analysis, which is performed in DM, can be used to obtain an embedding that is based on the coarse-grained process presented here, which results with an embedding of clusters to vectors (and not tensors).

The paper has the following structure. The problem setup is described in Section 2. Specifically, the DM method is discussed in Section 2.1 and the LDF method is discussed in Section 2.2. Section 3 introduces the localized diffusion process (LDP), which is the main construction in this paper. The pruning algorithm for constructing the LDP is presented in Section 3.1. Finally, the strong relation between LDF and LDP is presented in Section 3.2.

2. Problem setup

Let $X \subset \mathbb{R}^d$ be a dataset of n data-points that are sampled from a low-dimensional manifold that lies in a high-dimensional Euclidean ambient space. Assume the data consists of \hat{n} coherent disjoint clusters, which correspond to dense local neighborhoods that were generated by an affinity kernel. Assume that $C_1, C_2, \dots, C_{\hat{n}}$ are these clusters in the underlying manifold, where $X = \bigcup_{i=1}^{\hat{n}} C_i$ and $C_i \cap C_j = \emptyset$ for $i \neq j \in \{1, 2, \dots, \hat{n}\}$. Assume that

$$C : X \rightarrow \{C_1, C_2, \dots, C_{\hat{n}}\} \quad (2.1)$$

maps each data-point $x \in X$ to its cluster $C(x)$.

Remark about matrix notation. In this paper, we will deal with several matrices that represent relations between data-points or clusters of data-points. Let M be such a matrix where every row and column of M corresponds to a data-point in the dataset X or a subset of this dataset. It is convenient in this case to use the lowercase notation $m(x, y)$ to denote the cell in the x 's row and the y 's column in M . For $t \in \mathbb{Z}$, the notation $m^t(x, y)$ denotes cells in M^t , which is the t -th power of M . Similar notation will also be used for matrices with rows and columns that correspond to clusters of data-points.

2.1. Diffusion maps

The Diffusion Maps (DM) [2] methodology is based on constructing a Markovian diffusion process \mathcal{P} over a dataset. This process essentially defines random walks over data-points in the dataset. It consists of paths between these data-points, where each path $\mathcal{P} \in \mathcal{P}$ is a series of transitions (steps on the data-points), denoted by $\mathcal{P}_0 \rightarrow \mathcal{P}_1 \rightarrow \dots \rightarrow \mathcal{P}_\ell$, where $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_\ell \in X$, $\ell \geq 1$. Each path has a probability, which is defined by the probabilities of its transitions and will be discussed later. The length of the path \mathcal{P} , denoted by $\text{len}(\mathcal{P}) = \ell$, is its number of transitions. The source (i.e., the starting data-point) of the path is denoted by $s(\mathcal{P}) = \mathcal{P}_0$ and its destination is denoted by $t(\mathcal{P}) = \mathcal{P}_{\text{len}(\mathcal{P})} = \mathcal{P}_\ell$. When only paths of specific length $\ell = 1, 2, \dots$, are considered, the notation $\mathcal{P} \in \mathcal{P}^\ell$ will be used to denote that $\mathcal{P} \in \mathcal{P}$ and $\text{len}(\mathcal{P}) = \ell$. For example, a single transition in \mathcal{P} is a path of unit length $\mathcal{P} \in \mathcal{P}^1$.

In order to assign transition probabilities between data-points, an $n \times n$ affinity kernel K is defined on the dataset. Each cell $k(x, y)$, $x, y \in X$, in this kernel represents similarity, or proximity, between data-points. The kernel K is interpreted as both an affinity measure between data-points and a weighted adjacency matrix of a graph whose vertices are the data-points. It is assumed to satisfy the following properties:

- Each data-point has positive self-affinity: $k(x, x) > 0$, $x \in X$;
- Affinities are non-negative: $k(x, y) \geq 0$, $x, y \in X$;
- Affinities are symmetric: $k(x, y) = k(y, x)$, $x, y \in X$;
- The graph defined by the weighted adjacencies K is connected.

A popular affinity kernel is the isotropic Gaussian diffusion kernel $k(x, y) = \exp(-\|x - y\|/\varepsilon)$ with a suitable $\varepsilon > 0$. An alternative kernel, which is based on clustering patterns in the dataset, is the “shake-and-bake” kernel [1] that will be discussed in more details in Section 2.2.

Each data-point $x \in X$ corresponds to a vertex in the graph that is defined by K . The degree of a vertex in this graph is $q(x) \triangleq \sum_{y \in X} k(x, y)$, $x \in X$. The degree matrix Q is a diagonal matrix whose main diagonal holds these degrees (i.e., $q(x, x) = q(x)$ and $q(x, y) = 0$ for $x \neq y \in X$). Normalization by these degrees yields a row-stochastic matrix $P \triangleq Q^{-1}K$ that defines the transition probabilities

$$p(x, y) = \frac{k(x, y)}{q(x)}, \quad x, y \in X,$$

between data-points. These transition probabilities define the Markovian diffusion process \mathcal{P} over the dataset.

The diffusion process \mathcal{P} specifies the probability of “moving” from one data-point to another via paths of any given integer length $\ell \geq 1$. We denote this probability by

$$\Pr[x \xrightarrow{\mathcal{P}^\ell} y] \triangleq \Pr[t(\mathcal{P}) = y | s(\mathcal{P}) = x \wedge \mathcal{P} \in \mathcal{P}^\ell], \quad x, y \in X. \quad (2.2)$$

Since the diffusion process is a Markovian process with single-transition probabilities defined by P , Eq. (2.2) becomes

$$\Pr[x \xrightarrow{\mathcal{P}^\ell} y] = p^\ell(x, y), \quad x, y \in X, \ell = 1, 2, \dots,$$

where in particular $\Pr[x \xrightarrow{\mathcal{P}^1} y] = p(x, y)$.

The diffusion process \mathcal{P} is an ergodic Markov process. This means that \mathcal{P} has a stationary distribution in the limit $\ell \rightarrow \infty$ of the path lengths. Spectral analysis of this kernel yields a decaying spectrum $1 = \lambda_0 \geq |\lambda_1| \geq |\lambda_2| \geq \dots \geq 0$, where λ_i , $i = 0, 1, 2, \dots$, are the eigenvalues of P . When an isotropic Gaussian kernel is used, the decay of the spectrum can

be used to approximate the intrinsic dimension of the dataset's underlying manifold [2]. Dimensionality reduction can be achieved by spectral analysis of P [11] or, more conveniently, its symmetric conjugate $A = Q^{1/2} P Q^{-1/2}$ that is referred to as the diffusion affinity matrix. Let $\phi_0, \phi_1, \phi_2, \dots$ be the eigenvectors of A that correspond to the eigenvalues $\lambda_0, \lambda_1, \lambda_2, \dots$ (conjugation maintains the same eigenvalues of A), then DM is defined by the embedding

$$x \mapsto \Phi(x) \triangleq (|\lambda_0| \phi_0(x), |\lambda_1| \phi_1(x), |\lambda_2| \phi_2(x), \dots)^T, \quad x \in X.$$

A subset of these coordinates can be used by ignoring the eigenvectors with sufficiently small eigenvalues, which will anyway result with approximately-zero embedded coordinates.

A simple coarse-graining of the original diffusion process can be done by cluster pruning while defining transition probability between two clusters by considering all the paths between them. However, due to the decay of the diffusion kernel's spectrum, this method will converge fast (especially when applied several times) to the stationary distribution of the diffusion process. An alternative coarse-graining method, which excludes paths that are considered "global" from clusters point-of-view, will be presented in Section 3.

One aspect of any diffusion process coarse-graining is to translate a data-point terminology to a cluster terminology. This aspect must be addressed regardless of the paths that are considered when computing the transitional probabilities between clusters, since any path in the diffusion process is defined in terms of data-points. The probability of reaching every data-point on a path is determined by its starting data-point and by suitable powers of P . Since the clusters are disjoint, these probabilities can be easily interpreted as the probability of reaching a destination cluster. Specifically, it can be done by using the function C (Eq. (2.1)) and by summing the appropriate probabilities. Paths that start in a source cluster, denoted by $s(\mathcal{P}) \in C_i$, $\mathcal{P} \in \mathcal{P}$, $i = 1, \dots, \hat{n}$, require a non-trivial interpretation in terms of a source data-point $s(\mathcal{P}) = x \in C_i$. This interpretation should be defined by probability terms.

We will use the same intuition that was used to construct the transitional probability matrix P in order to define the probability $\Pr[s(\mathcal{P}) = x \in C_i | s(\mathcal{P}) \in C_i]$, $i = 1, \dots, \hat{n}$, $\mathcal{P} \in \mathcal{P}$. The kernel K was interpreted as weighted adjacencies of a graph whose vertices are the data-points in X . According to this interpretation, the degree of each data-point $x \in X$ is a sum of the edges weights $k(x, y)$, $y \in X$ that begin at x . To measure the occurrence probability of the transition $x \rightarrow y$ when starting at x , the weight of the edge (x, y) is divided by the total weight of the edges starting at x , which gives the probability measure $p(x, y) = k(x, y)/q(x)$. Assume the volume of the cluster C_i , $i = 1, \dots, \hat{n}$, is defined as $\text{vol}(C_i) \triangleq \sum_{x \in C_i} q(x)$. Therefore, the volume of a cluster is the total sum of the degrees of the data-points in this cluster, which is the sum of the weights of all the edges that start in C_i . According to the same reasoning as before, the occurrence probability of the transition $x \rightarrow y$, $x \in C_i$, $y \in X$, which started at the cluster C_i , is $\frac{k(x, y)}{\text{vol}(C_i)}$. Therefore, the transition probability, which starts at C_i to actually starts at a specific data-point $x \in C_i$, is

$$\Pr[s(\mathcal{P}) = x \in C_i | s(\mathcal{P}) \in C_i] = \sum_{y \in X} \frac{k(x, y)}{\text{vol}(C_i)} = \frac{q(x)}{\text{vol}(C_i)}, \quad (2.3)$$

because the transitions to different designated data-points are independent events. Notice that the choice of the first transition in a path is independent of its length. Thus, the presented probability is independent of the length of the path \mathcal{P} and the assumption that $\mathcal{P} \in \mathcal{P}^\ell$ for some $\ell \geq 1$ does not affect it.

2.2. Localized Diffusion Folders (LDF)

As described in the DM brief overview in Section 2.1, P is the affinity matrix of the dataset and it is used to find the diffusion distances between data-points. This distance metric can be used to cluster data-points according to the diffusion distances propagation that is controlled by the time parameter t . In addition, it can be used to construct a bottom-up hierarchical data clustering. For $t = 1$, the affinity matrix reflects direct connections between data-points. These connections can be interpreted as local adjacencies between data-points. The resulting clusters preserve the local neighborhood of each data-point. These clusters are the bottom level in the hierarchy. By raising t , which means time advancement, the affinity matrix is changed accordingly and it reflects indirect rare connections between data-points in the graph. The diffusion distance between data-points in the graph accounts for all possible paths of length t between these data-points at a given time step. The more we advance in time the more we increase indirect and global connections. Therefore, by raising t we can construct the upper levels of the clustering hierarchy. In each time step, it is possible to merge more and more low-level clusters since there are more and more new paths between them. The resulting clusters reflect global neighborhood of each data-point that is highly affected by the advances of the parameter t .

The major risk in this global approach is that increasing t will also increase noise, which is classified as connections between data-points that are not closely related in the affinity matrix. Moreover, clustering errors in the lower levels of the hierarchy will diffuse to the upper levels of the hierarchy and hence will significantly affect the correctness of the upper levels clustering. As a result, some areas in the graph, which are assumed to be separated, will be connected by the new noise-result and error-result paths. Thus, erroneous clusters will be generated (a detailed description of this situation is given in [1]). This type of noise significantly affects the diffusion process and eventually the resulting clusters will not reflect the correct relations among the data-points. Although these clusters consist of data-points that are adjacent according to

their diffusion distances, the connections among these data-points in each cluster can be classified as too global and too loose that generate inaccurate clusters.

A hierarchical clustering method of high-dimensional data via the *Localized Diffusion Folders* (LDF) methodology is introduced in [1]. This methodology overcomes the problems that were described above. It is based on the key idea that clustering of data-points should be achieved by utilizing the local geometry of the data and the by local neighborhood of each data-point and by constructing a new local geometry every advance in time. The new geometry is constructed according to local connections and according to diffusion distances in previous time steps. This way, as we advance in time, the geometry from the induced affinity reflects better the data locality while the “affinity noise” in the new localized matrix decreases and the accuracy of the resulting clusters is improved.

LDF is introduced to achieve the described local geometry and to preserve it along the hierarchical construction. The LDF framework provides a multi-level partitioning (similar to Voronoi diagrams in diffusion metric) of the data into local neighborhoods that are initiated by several random selections of data-points or folders of data-points in the diffusion graph and by defining local diffusion distances between them. Since every different selection of initial data-points yields a different set of *Diffusion Folders* (DF), it is crucial to repeat this selection process several times. The multiple system of folders, which we get at the end of this random selection process, defines a new affinity and this reveals a new geometry in the graph. This localized affinity is a result of what is called the “shake & bake” process in [1]. First, we “shake” the multiple Voronoi diagrams together in order to get rid of the noise in the original affinity. Then, we “bake” a new cleaner affinity that is based on the actual geometry of the data while eliminating rare connections between data-points. This affinity is more accurate than the original affinity since instead of defining a general affinity on the graph, we let the data define its localized affinity on the graph.

In every time step, this multi-level partitioning defines a new localized geometry of the data and a new localized affinity matrix that is used in the next time step. In every time step, we use the localized geometry and the LDF that were generated in the previous time step to define the localized affinity between DF. The affinity between two DF is defined by the localized diffusion distance metric between data-points in the two DF. In order to define this distance between these DF, we construct a local sub-matrix that contains only the affinities between data-points (or between DF) of the two DF. This sub-matrix is raised to the power of the current time step (according to the current level in the hierarchy) and then it is used to find the localized diffusion distance between the two DF.

The result of this clustering method is a bottom-up hierarchical data clustering where each level in the hierarchy contains DF of DF from lower levels. Each level in the hierarchy defines a new localized affinity (geometry) that is dynamically constructed and it is used by the upper level. This methodology preserves the local neighborhood of each data-point while eliminating the noisy connections between distinct points and areas in the graph.

In summary, [1] deals with new methodologies to denoise empirical graphs. Usually, in applications data is connected through spurious connections. One of the goals of [1] is to introduce a notion of consistency of connections in order to repair a noisy network. This consistency is achieved through the construction of a forest of partition trees, which redefine the connectivity in the network. This opens the door to robust processing of data clouds in which group consistency is exploited.

3. Localized diffusion process

In this section, we present a coarse-graining diffusion process between clusters in a dataset. The transitions between clusters, which are considered as vertices in this process, will be defined by certain paths in the original diffusion process. Definition 3.1 introduces the notion of a localized path, which will be used to define these transitions. Then, in Definition 3.3, these localized paths will be used to define the localized diffusion process between clusters.

Definition 3.1 (*Localized ℓ -path*). A localized ℓ -path in a diffusion process \mathcal{P} is the path $\mathcal{P} \in \mathcal{P}^\ell$ of length ℓ that traverses solely through data-points in its source and destination clusters, i.e., $\mathcal{P}_0, \mathcal{P}_1, \dots, \mathcal{P}_\ell \in C(s(\mathcal{P})) \cup C(t(\mathcal{P}))$.

The difference between localized and non-localized paths is demonstrated in Fig. 3.1. The path in Fig. 3.1(a) traverses through data-points in its source cluster, then passes via a single transition to its destination cluster and then traverses in it to its destination data-point. Therefore, it does not pass through any cluster other than its source and destination clusters and thus it is localized. On the other hand, the non-localized path in Fig. 3.1(b), traverses through a third intermediary cluster, thus it is not localized.

Notice that a localized path does not necessarily contain a single transition between its source and destination clusters. Fig. 3.2 illustrates two such non-trivial paths. A path that traverses solely in a single cluster (see Fig. 3.2(a)) is in fact a localized path from the cluster to itself. A localized path can also alternate between its source and destination clusters a few times before reaching its final destination, as shown in Fig. 3.2(b). As long as the cluster involves only its source and destination cluster/s (whether they are identical or not) without passing through any intermediary cluster, then it is considered to be localized.

We denote the set of all localized ℓ -paths in the diffusion process \mathcal{P} by $\mathcal{L}(\mathcal{P}^\ell) \subseteq \mathcal{P}^\ell$. The usual diffusion transition probabilities between data-points in a dataset via paths of a given length $\ell \geq 1$ were described in Section 2.1. These probabilities consider all the paths of length ℓ between two data-points. The construction presented in this paper only considers

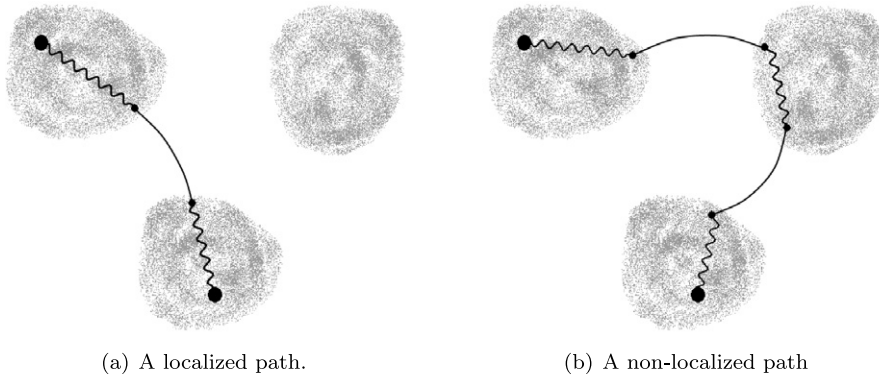


Fig. 3.1. Illustration of the difference between localized and non-localized paths.

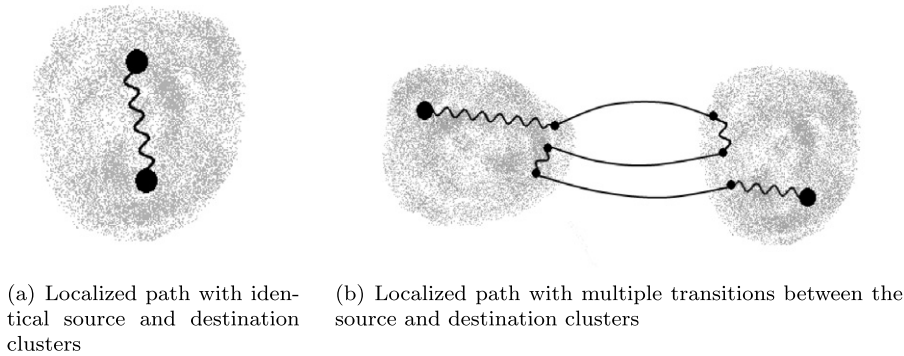


Fig. 3.2. Illustration of non-trivial localized paths.

localized paths and ignores other paths, which are considered “global” from a cluster point-of-view. Therefore, we define the localized transition probabilities, which describe the probabilities of a transition from $x \in C_i$ to $y \in C_j$, $i, j = 1, \dots, \hat{n}$, via localized ℓ -paths, as

$$\Pr[x \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} y] \triangleq \Pr[t(\mathcal{P}) = y \wedge \mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) = x \wedge \mathcal{P} \in \mathcal{P}^\ell]. \quad (3.1)$$

Similarly, the localized transition probability from $x \in C_i$ to the cluster C_j is defined as

$$\Pr[x \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j] \triangleq \Pr[t(\mathcal{P}) \in C_j \wedge \mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) = x \wedge \mathcal{P} \in \mathcal{P}^\ell]. \quad (3.2)$$

Finally, the localized transition probability from the cluster C_i to the cluster C_j is defined as

$$\Pr[C_i \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j] \triangleq \Pr[t(\mathcal{P}) \in C_j \wedge \mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{P}^\ell]. \quad (3.3)$$

The original transition probabilities are clearly related to the diffusion operator \mathcal{P} via Eq. (2.2). The defined localized transition probabilities do not have such a direct relation with the diffusion operator. They will be further explored in Section 3.1.

The localized transition probabilities in Eq. (3.3) consider localized paths from a source cluster to a destination cluster. Not all the paths from the source cluster are localized. Therefore, only a portion of the paths from a given cluster (to any other cluster) are actually viable for consideration with these probabilities. Definition 3.2 provides a measure for the portion of viable paths going out from a cluster from all the paths starting in it.

Definition 3.2 (ℓ -Path localization probability). The ℓ -path localization probability (lpr) of a cluster C_i , $i = 1, \dots, \hat{n}$, is

$$\text{lpr}^\ell(C_i) \triangleq \Pr[\mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{P}^\ell].$$

It is the probability that a path of length ℓ , which starts at this cluster, is a localized path.

Definition 3.3 uses the defined localized paths in the original diffusion process to define a localized diffusion process between clusters.

Definition 3.3 (ℓ -Path localized diffusion process). Let \mathcal{P} be a diffusion (random walk) process defined on the data-points of the dataset X . An ℓ -path localized diffusion process $\widehat{\mathcal{P}}$ is a random walk on the clusters $C_1, C_2, \dots, C_{\hat{n}}$ where a transition from C_i to C_j , $i, j = 1, \dots, \hat{n}$, represents all the localized ℓ -paths in the diffusion process \mathcal{P} from data-points in C_i to data-points in C_j . The probability of such a transition, according to \mathcal{P} , is the probability to reach the destination cluster C_j when starting at the source cluster C_i and traveling solely via localized ℓ -paths.

The ℓ -path localized diffusion process is a Markovian random-walk process. Thus, its transition probabilities are completely governed by its single-step transition probabilities. These probabilities can be computed, by definition, according to the transition probabilities of the original diffusion process. Using notations similar to the ones used for the original diffusion process, we get the single-step transition probabilities

$$\Pr[C_i \xrightarrow{\widehat{\mathcal{P}}^1} C_j] \triangleq \Pr[t(\mathcal{P}) \in C_j | s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell)], \quad i, j = 1, \dots, \hat{n}, \quad (3.4)$$

for the ℓ -path localized diffusion process $\widehat{\mathcal{P}}$. Notice that these differ from the probabilities in Eq. (3.3). The former considers the term $\mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell)$ in the hypothesis part, since it considers only the localized paths of the original diffusion process. The latter considers this term in the condition part, since it computes the probability over all the paths in the original diffusion.

Ergodicity is one of the main properties in the diffusion process that is used by DM [2]. Ergodicity means that the eigenvalues of P have a magnitude of at most one, and therefore its spectrum decays with time as a function of the numerical rank of the transitional kernel. As we advance the diffusion process in time, it converges to a stationary distribution and therefore its long term state can be represented by a low-dimensional space. Proposition 3.1 shows that the coarse-graining suggested here preserves this property, i.e., the ℓ -path localized diffusion process is ergodic and its transition matrix has a decaying spectrum.

Proposition 3.1. *The localized diffusion process $\widehat{\mathcal{P}}$, which is defined by Definition 3.3, is an ergodic Markov process.*

Proof. According to [2], the original diffusion process \mathcal{P} is aperiodic and irreducible. We will show that $\widehat{\mathcal{P}}$ is also aperiodic and irreducible process. The ergodicity follows from these properties.

From the aperiodicity of \mathcal{P} we have $p(x, x) > 0$ for every $x \in X$. Let $\mathcal{P} \in \mathcal{P}^\ell$ be a path with $\mathcal{P}_i = x$, $i = 0, \dots, \ell$. Obviously, this path is a localized ℓ -path from $C(x)$ to itself. The probability of this path is $(p(x, x))^\ell > 0$. Therefore, the transition probability $\Pr[C(x) \xrightarrow{\widehat{\mathcal{P}}^\ell} C(x)]$, which sums the probabilities of all the localized ℓ -paths from $C(x)$ to itself, must be nonzero. This argument holds for every $x \in C_i \subseteq X$, $i = 1, \dots, \hat{n}$, and thus holds for every cluster $C_i = C(x)$. Therefore, the process $\widehat{\mathcal{P}}$ is aperiodic.

Due to the irreducibility of the original diffusion process \mathcal{P} , there exists a path $\mathcal{P} \in \mathcal{P}$ with nonzero probability between every pair of data-points $x \neq y \in X$. For each transition $\mathcal{P}_i \rightarrow \mathcal{P}_{i+1}$, $i = 0, \dots, \text{len}(\mathcal{P}) - 1$, in this path, the following localized ℓ -path

$$\mathcal{P}' = \underbrace{\mathcal{P}_i \rightarrow \mathcal{P}_i \rightarrow \dots \rightarrow \mathcal{P}_i}_{\ell-1 \text{ transitions}} \rightarrow \mathcal{P}_{i+1}$$

between $C(\mathcal{P}_i)$ and $C(\mathcal{P}_{i+1})$ is constructed. Due to the aperiodicity of \mathcal{P} , the first $\ell - 1$ transitions have nonzero probability. The last transition of \mathcal{P}' is the same transition $\mathcal{P}_i \rightarrow \mathcal{P}_{i+1}$ from \mathcal{P} , which also has nonzero probability. Therefore, the path \mathcal{P}' is a localized ℓ -path between $C(\mathcal{P}_i)$ and $C(\mathcal{P}_{i+1})$ with nonzero probability. This holds for every transition in \mathcal{P} and thus the transition probability from $C(\mathcal{P}_i)$ to $C(\mathcal{P}_{i+1})$ via the localized ℓ -paths is nonzero for each $i = 0, \dots, \text{len}(\mathcal{P}) - 1$. Thus, the path

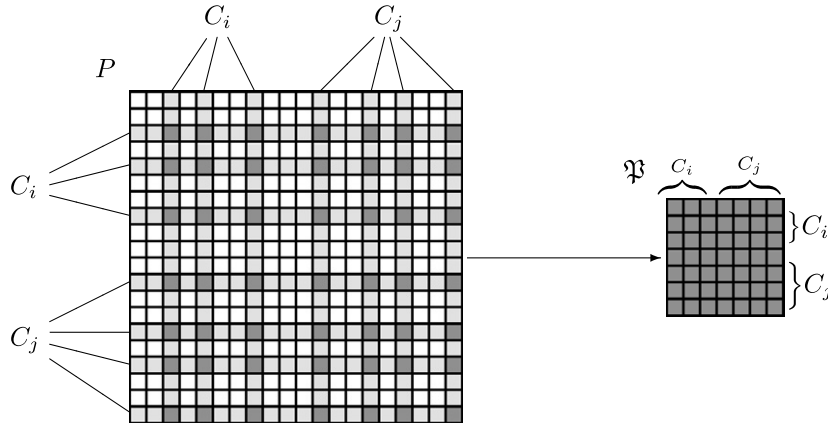
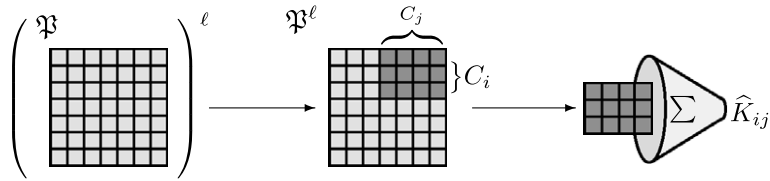
$$C(x) = C(\mathcal{P}_0) \rightarrow C(\mathcal{P}_1) \rightarrow \dots \rightarrow C(\mathcal{P}_{\text{len}(\mathcal{P})}) = C(y)$$

has nonzero probability in $\widehat{\mathcal{P}}$. Since x, y were chosen arbitrarily, this holds for every pair of clusters and thus $\widehat{\mathcal{P}}$ is irreducible. Together with the aperiodicity that was shown above, $\widehat{\mathcal{P}}$ is ergodic. \square

In this section, we introduced a coarse-grained diffusion process (i.e., the ℓ -path localized diffusion process) that preserves the crucial properties of the DM. A coarse-graining algorithm, which constructs this process, is presented in Section 3.1. In Section 3.2, we will show that this process is directly related to the construction presented in [1]. Specifically, the presented coarse-graining is related to the pruning done at the transition between levels in the LDF hierarchy.

3.1. Pruning algorithm

The ℓ -path localized diffusion process described in Section 3 is a coarse-grained version of the original diffusion process. As a Markovian process, it defines a transition probability matrix between clusters. Algorithm 3.1 shows how to construct this transition probability matrix, denoted by \widehat{P} , based on the transition probability matrix of the original diffusion process. In addition to \widehat{P} , the algorithm outputs the degree matrix \widehat{Q} that holds the degrees of the clusters on its diagonal. Theorem 3.2 shows that the resulting matrix \widehat{P} defines a localized diffusion process.

(a) Construction of the submatrix \mathfrak{P} from the matrix P (b) Computation of \hat{K}_{ij} as a weighted sum of cells in \mathfrak{P}^ℓ **Fig. 3.3.** Construction of the pruned kernel \hat{K}_{ij} between the clusters C_i and C_j .

Algorithm 3.1 performs a coarse-graining of the original diffusion process by pruning the clusters into vertices of a Markovian random-walk process. The transition probabilities of this process are determined by the row-stochastic transition matrix \hat{P} . For each pair of clusters, C_i and C_j , the algorithm considers the sub-matrix \mathfrak{P} of P , which contains only rows and columns of data-points in $C_i \cup C_j$ (see Fig. 3.3(a)). The algorithm then calculates the affinity, denoted by K_{ij} , between the clusters C_i and C_j . First, it raises the sub-matrix \mathfrak{P} to the ℓ -th power in order to generate the ℓ -path localized transition probabilities between points in C_i and C_j . Then, the affinity between these clusters is a weighted sum of the elements in \mathfrak{P}^ℓ , where the weight of the element $p^\ell(x, y)$ is $q(x)$ (see Fig. 3.3(b)). Finally, the degree of each cluster is calculated by summing its affinities $\hat{Q}_{ii} = \sum_{j=1}^{\hat{n}} \hat{K}_{ij}$ with all the clusters.

Notice that Algorithm 3.1 is similar to the pruning algorithm described in [1, Section 3.3]. Both algorithms get an input matrix of relations between data-points and a clustering function that assigns each point to its cluster. Then, for each pair

Algorithm 3.1: Transition matrix pruning.

Input: Dataset X of n data-points;
Clustering function $C : X \rightarrow \{C_1, C_2, \dots, C_{\hat{n}}\}$ of the data into \hat{n} clusters;
Transition probability matrix P between data-points in X ;
Parameter ℓ of the path length.
Output: A row-stochastic $\hat{n} \times \hat{n}$ matrix \hat{P} that represents transitions between clusters;
A diagonal matrix \hat{Q} that contains the degrees of the clusters on its diagonal.
foreach $i, j = 1, \dots, \hat{n}$ **do**
 $\mathfrak{P} \leftarrow |C_i \cup C_j| \times |C_i \cup C_j|$ square matrix;
 // Denote by $p(x, y)$ the cell of \mathfrak{P} in the row of x
 // and the column of y ($x, y \in C_i \cup C_j$).
 // Denote by $p^\ell(x, y)$ the same cell in \mathfrak{P}^ℓ .
 foreach $x, y \in C_i \cup C_j$ **do**
 $p(x, y) \leftarrow p(x, y)$;
 end
 $\hat{K}_{ij} \leftarrow \sum_{x \in C_i, y \in C_j} q(x) p^\ell(x, y)$;
end
foreach $i = 1, \dots, \hat{n}$ **do**
 $\hat{Q}_{ii} \leftarrow \sum_{j=1}^{\hat{n}} \hat{K}_{ij}$;
end
 $\hat{P} \leftarrow \hat{Q}^{-1} \hat{K}$;

of clusters, these algorithms consider a sub-matrix that contains the relations between the data-points in the two considered clusters. In order to achieve a scalar representation of the relation between the considered clusters, both algorithms aggregate the elements of a suitable power of the considered sub-matrix.

However, these algorithms differ in the input matrix itself and in the aggregation function that is being used. Algorithm 3.1 gets a transition probability matrix as an input and uses a weighted sum for the aggregation, while the algorithm in [1, Section 3.3] gets an affinity matrix as an input and suggests three different aggregations of the sub-matrix elements.

Theorem 3.2 shows that the resulting Markov process in Algorithm 3.1 is in fact a transition probability matrix of the ℓ -path localized diffusion process that was defined in Definition 3.3.

Theorem 3.2. *The output matrix \widehat{P} from Algorithm 3.1 is a transition probability matrix of an ℓ -path localized diffusion process.*

In order to prove Theorem 3.2, we need Lemmas 3.3 and 3.4 that relate the matrices used in Algorithm 3.1 to the original diffusion process.

Lemma 3.3. *Let \mathfrak{P} be the sub-matrix of P defined in a single iteration of Algorithm 3.1 for specific $i, j = 1, \dots, \hat{n}$. \mathfrak{P} is related to the localized transition probabilities of \mathcal{P} in the following ways:*

1. $p^\ell(x, y) = \Pr[x \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} y];$
2. $\sum_{y \in C_j} p^\ell(x, y) = \Pr[x \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j];$
3. $\sum_{x \in C_i} \sum_{y \in C_j} \frac{q(x)}{\text{vol}(C_i)} p^\ell(x, y) = \Pr[C_i \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j].$

The proof of Lemma 3.3 is given in Appendix A.

Lemma 3.4. *Let $\hat{q}(C_i) \triangleq \widehat{Q}_{ii} \triangleq \sum_{j=1}^{\hat{n}} \widehat{K}_{ij}$, $i = 1, \dots, \hat{n}$, be the degree (i.e., row sum) defined in Algorithm 3.1. Then, $\hat{q}(C_i) = \text{vol}(C_i) \cdot \text{lpr}^\ell(C_i)$.*

Proof. According to Definition 3.2

$$\begin{aligned} \text{lpr}^\ell(C_i) &= \Pr[p \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{P}^\ell] \\ &= \sum_{j=1}^{\hat{n}} \Pr[C_i \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j], \quad i = 1, \dots, \hat{n}. \end{aligned}$$

Combining with property (3) of Lemma 3.3 yields

$$\text{lpr}^\ell(C_i) = \sum_{j=1}^{\hat{n}} \sum_{x \in C_i} \sum_{y \in C_j} \frac{q(x)}{\text{vol}(C_i)} p^\ell(x, y), \quad i = 1, \dots, \hat{n},$$

where \mathfrak{P} depends on the choice of i and j . By using the matrix \widehat{K} from Algorithm 3.1 we get

$$\text{lpr}^\ell(C_i) = \sum_{j=1}^{\hat{n}} \frac{K_{ij}}{\text{vol}(C_i)} = \frac{\hat{q}(C_i)}{\text{vol}(C_i)}, \quad i = 1, \dots, \hat{n},$$

and multiplying by $\text{vol}(C_i)$ yields the desired result. \square

Lemmas 3.3 and 3.4 relate the localized transition probabilities from the diffusion process to the original diffusion transition probabilities via the matrices constructed in Algorithm 3.1. These relations can now be used to prove Theorem 3.2.

Proof of Theorem 3.2. Consider two clusters C_i and C_j , $i, j = 1, \dots, \hat{n}$. According to Algorithm 3.1, $\widehat{P}_{ij} \triangleq \frac{\widehat{K}_{ij}}{Q_{ii}}$ and $\widehat{K}_{ij} \triangleq \sum_{x \in C_i, y \in C_j} q(x) p^\ell(x, y)$. Using Lemmas 3.3 and 3.4 we obtain

$$\widehat{P}_{ij} = \frac{\text{vol}(C_i) \cdot \Pr[C_i \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j]}{\text{vol}(C_i) \cdot \text{lpr}^\ell(C_i)} = \frac{\Pr[C_i \xrightarrow{\mathcal{L}(\mathcal{P}^\ell)} C_j]}{\text{lpr}^\ell(C_i)}.$$

By Definition 3.2 and Eq. (3.3),

$$\hat{P}_{ij} = \frac{\Pr[t(\mathcal{P}) \in C_j \wedge \mathcal{P} \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{P}^\ell]}{\Pr[p \in \mathcal{L}(\mathcal{P}^\ell) | s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{P}^\ell]},$$

and by conditional probability considerations we get

$$\hat{P}_{ij} = \Pr[t(\mathcal{P}) \in C_j | p \in \mathcal{L}(\mathcal{P}^\ell) \wedge s(\mathcal{P}) \in C_i \wedge \mathcal{P} \in \mathcal{P}^\ell]. \quad (3.5)$$

The term $\mathcal{P} \in \mathcal{P}^\ell$ in the hypothesis of Eq. (3.5) is redundant by the localized ℓ -path. Thus, by combining with Eq. (3.4) we get $\hat{P}_{ij} = \Pr[C_i \xrightarrow{\hat{P}^1} C_j]$ and the theorem is proved. \square

3.2. Relation to LDF

In the original diffusion, it is assured that the magnitude of the eigenvalues of P is between zero and one. Another important property of P is the existence of a symmetric conjugate A . Being a symmetric matrix, the eigenvalues of A are all real and its left and right eigenvectors are identical. The matrix A has the same eigenvalues as P and its eigenvectors are related to those of P by the same conjugation that relates A to P . The additional information provided by the symmetric conjugate A allows for a simple spectral analysis to be used to achieve dimensionality reduction as described in [2]. Theorem 3.5 shows that these properties also apply to the ergodic ℓ -path localized diffusion process \hat{P} .

Theorem 3.5. *Let \hat{P} be the transition probability matrix of a localized ℓ -path diffusion process, which resulted from Algorithm 3.1. Let \hat{Q} be the corresponding degree matrix. Then the conjugate matrix $\hat{A} = \hat{Q}^{1/2} \hat{P} \hat{Q}^{-1/2}$ is symmetric. Furthermore, \hat{A} is equivalent to the result from the weighted-sum LDF runner in [1, Section 3.3].*

Proof. Consider two clusters C_i and C_j , $i, j = 1, \dots, \hat{n}$. Let \mathfrak{P} be the matrix defined for them in the corresponding iteration of Algorithm 3.1. Let Ω be a diagonal $|C_i \cup C_j| \times |C_i \cup C_j|$ matrix where each cell on its diagonal corresponds to a data-point $x \in C_i \cup C_j$ and holds its degree $q(x) = q(x)$. As discussed in Section 2.1, the diffusion affinity matrix $A = Q^{1/2} P Q^{-1/2}$ is a symmetric conjugate of the diffusion operator P . Its cells are

$$a(x, y) = \sqrt{\frac{q(x)}{q(y)}} \cdot p(x, y), \quad x, y \in X.$$

Let $\mathfrak{A} = \Omega^{1/2} \mathfrak{P} \Omega^{-1/2}$ be a $|C_i \cup C_j| \times |C_i \cup C_j|$ conjugate of \mathfrak{P} , then its cells are

$$\alpha(x, y) = \sqrt{\frac{q(x)}{q(y)}} \cdot p(x, y) = \sqrt{\frac{q(x)}{q(y)}} \cdot p(x, y) = a(x, y), \quad x, y \in C_i \cup C_j. \quad (3.6)$$

From the symmetry of A , we get $\alpha(x, y) = \alpha(y, x)$, $x, y \in C_i \cup C_j$ and \mathfrak{A} is symmetric. The powers of a symmetric matrix are also symmetric, thus \mathfrak{A}^ℓ , $\ell \geq 1$, which was given as a parameter to Algorithm 3.1, is also symmetric and since the terms $\Omega^{-1/2} \Omega^{1/2} = I$ are canceled, then

$$\mathfrak{A}^\ell = \underbrace{\Omega^{1/2} \mathfrak{P} \Omega^{-1/2} \cdot \Omega^{1/2} \mathfrak{P} \Omega^{-1/2} \dots \Omega^{1/2} \mathfrak{P} \Omega^{-1/2}}_{\ell \text{ times}} = \Omega^{1/2} \mathfrak{P}^\ell \Omega^{-1/2}. \quad (3.7)$$

The symmetry is maintained by multiplying the symmetric matrix \mathfrak{A}^ℓ from left and right by the diagonal matrix $\Omega^{1/2}$. Thus the resulting matrix is

$$\Omega^{1/2} \mathfrak{A}^\ell \Omega^{1/2} = \Omega^{1/2} \Omega^{1/2} \mathfrak{P}^\ell \Omega^{-1/2} \Omega^{1/2} = \Omega \mathfrak{P}^\ell.$$

According to Algorithm 3.1 and since $q(x) = q(x)$ for $x \in C_i \cup C_j$, then

$$\hat{K}_{ij} = \sum_{x \in C_i, y \in C_j} q(x) p^\ell(x, y).$$

According to the symmetry of $\Omega \mathfrak{P}^\ell$, we obtain

$$\hat{K}_{ij} = \sum_{x \in C_i} \sum_{y \in C_j} q(y) p^\ell(y, x).$$

The same matrices \mathfrak{P} and Ω are also used in the iteration that computes \hat{K}_{ji} in Algorithm 3.1, thus

$$\hat{K}_{ji} = \sum_{y \in C_j} \sum_{x \in C_i} q(y) p^\ell(y, x) = \hat{K}_{ij}$$

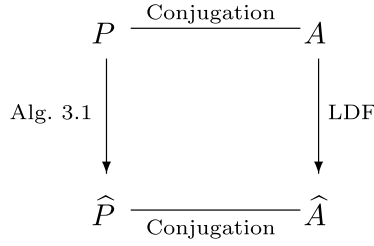


Fig. 3.4. The relation between Algorithm 3.1 and the LDF pruning algorithm.

holds and \hat{K} is symmetric. Since $\hat{P} \triangleq \hat{Q}^{-1}\hat{K}$ by Algorithm 3.1, then

$$\hat{A} = \hat{Q}^{1/2}\hat{P}\hat{Q}^{-1/2} = \hat{Q}^{-1/2}\hat{K}\hat{Q}^{-1/2}. \quad (3.8)$$

Multiplication by the diagonal matrix $\hat{Q}^{-1/2}$ from both sides maintains the symmetry of \hat{K} , thus \hat{A} is also symmetric. Combining Eq. (3.8) and the definition of \hat{K} in Algorithm 3.1, yields

$$\hat{A}_{ij} = \frac{\hat{K}_{ij}}{\sqrt{\hat{Q}_{ii}}\sqrt{\hat{Q}_{jj}}} = \sum_{x \in C_i} \sum_{y \in C_j} \frac{q(x)p^\ell(x, y)}{\sqrt{\hat{q}(C(x))\hat{q}(C(y))}}.$$

Together with Eq. (3.7), we receive

$$\hat{A}_{ij} = \sum_{x \in C_i} \sum_{y \in C_j} \frac{\sqrt{q(x)}\sqrt{q(y)}\alpha^\ell(x, y)}{\sqrt{\hat{q}(C(x))\hat{q}(C(y))}}.$$

Let

$$w_{xy} \triangleq \sqrt{\frac{q(x)q(y)}{\hat{q}(C(x))\hat{q}(C(y))}}, \quad x, y \in X,$$

then the following weighted sum is obtained:

$$\hat{A}_{ij} = \sum_{x \in C_i} \sum_{y \in C_j} w_{xy}\alpha^\ell(x, y).$$

Finally, according to Eq. (3.6), the matrix \mathfrak{A} is a sub-matrix of A , which contains cells in rows and columns that correspond to data-points in $C_i \cup C_j$. This is exactly the sub-matrix used in the corresponding iteration (for C_i and C_j) in the LDF algorithm [1, Section 3.3], and thus \hat{A}_{ij} contains a weighted sum of the cells that are combined by the LDF runners [1, Section 3.3]. Therefore, the matrix \hat{A} , which is a symmetric conjugate of \hat{P} , can be directly obtained by a weighted-sum LDF runner with the defined weights w_{xy} , $x, y \in X$. \square

From Theorem 3.5, the symmetric matrix \hat{A} can be used for spectral analysis of the localized diffusion since it has the same spectrum as \hat{P} and its eigenvectors are related to the eigenvectors of \hat{P} by the same conjugation that relates \hat{A} to \hat{P} . In fact, \hat{A} is a result of the LDF runner, which is used to prune a level in the LDF hierarchy to the next (higher) level, and thus, it can be constructed directly from A without using P , \hat{P} and the conjugations (see Fig. 3.4).

If we denote the eigenvalues of \hat{A} by $1 = \hat{\lambda}_0 \geq |\hat{\lambda}_1| \geq |\hat{\lambda}_2| \geq \dots$ and the corresponding eigenvectors by $\hat{\phi}_0, \hat{\phi}_1, \hat{\phi}_2, \dots$, we can define a coarse-grained DM, which we call the ℓ -path Localized Diffusion Map (LDM). This map embeds each cluster C_i , $i = 1, \dots, \hat{n}$, to a point

$$\hat{\Phi}(C_i) = (|\hat{\lambda}_0|\hat{\phi}_0(C_i), |\hat{\lambda}_1|\hat{\phi}_1(C_i), |\hat{\lambda}_2|\hat{\phi}_2(C_i), \dots)^T.$$

According to the above discussion, this embedding has the same properties as the DM embedding, which was presented in [2].

By combining the original DM, which embeds data-points, and the presented LDM, which embeds clusters, we obtain a two-level embedding (i.e., a data-point level and a cluster level). Moreover, the LDM is defined by spectral analysis of the affinity constructed by LDF. Therefore, it can be defined for each level of the LDF hierarchy. Thus, we get a multilevel embedding of the data where, in each level, the corresponding DF are embedded. Furthermore, each coarse-graining iteration (between LDF levels) prunes longer paths to single transitions, and thus, a wider time scale of the diffusion is considered. Therefore, the multilevel embedding, which results from the LDM and from the LDF hierarchy, provides a multiscale coarse-grained DM.

4. Conclusion

The presented ℓ -path localized diffusion process introduces a coarse-grained version of the diffusion process that is used in DM for high-dimensional data analysis and dimensionality reduction. This coarse-grained process preserves the locality of the data by pruning previously detected clusters while considering only localized paths between them. A simple pruning algorithm can be used to perform the described coarse-graining while maintaining the essential algebraic and spectral properties of the DM process as was introduced in [2]. Furthermore, this pruning is equivalent (via conjugation) to the one performed by the LDF algorithm when it computes the LDF hierarchy. By combining the results of this paper with the ones in [1], the LDF hierarchy is shown to provide the foundations for a multi-scale coarse-grained DM-based embedding of data-points and clusters/folders to a low-dimensional space.

Acknowledgments

This research was partially supported by the Israel Science Foundation (Grant No. 1041/10). The first author was also supported by the Eshkol Fellowship from the Israeli Ministry of Science & Technology.

Appendix A. Proof of Lemma 3.3

Proof. Since \mathcal{P} is a Markovian random-walk process with a transition probability matrix P , then the probability of a path $\mathcal{P} \in \mathcal{P}^\ell$ is $\prod_{\xi=1}^{\ell} p(\mathcal{P}_{\xi-1}, \mathcal{P}_\xi)$. The probability $\Pr[x \xrightarrow{\mathcal{P}^\ell} y]$, which is defined in Eq. (3.1), considers only paths with $\mathcal{P}_0 = s(\mathcal{P}) = x$, $\mathcal{P}_\ell = t(\mathcal{P}) = y$ and $\mathcal{P}_\xi \in C_i \cup C_j$, $\xi = 1, \dots, \ell - 1$, thus

$$\Pr[x \xrightarrow{\mathcal{P}^\ell} y] = \sum_{\mathcal{P}_1, \dots, \mathcal{P}_{\ell-1} \in C_i \cup C_j} \left[p(x, \mathcal{P}_1) \cdot \prod_{\xi=2}^{\ell-1} p(\mathcal{P}_{\xi-1}, \mathcal{P}_\xi) \cdot p(\mathcal{P}_{\ell-1}, y) \right], \quad x \in C_i, y \in C_j.$$

By Algorithm 3.1, $p(\mathcal{P}_{\xi-1}, \mathcal{P}_\xi) = p(\mathcal{P}_{\xi-1}, \mathcal{P}_\xi)$, $\xi = 1, \dots, \ell$ when $\mathcal{P}_1, \dots, \mathcal{P}_{\ell-1} \in C_i \cup C_j$, $\mathcal{P}_0 = x \in C_i$ and $\mathcal{P}_\ell = y \in C_j$. Therefore,

$$\begin{aligned} \Pr[x \xrightarrow{\mathcal{P}^\ell} y] &= \sum_{\mathcal{P}_1, \dots, \mathcal{P}_{\ell-1} \in C_i \cup C_j} \left[p(x, \mathcal{P}_1) \cdot \prod_{\xi=2}^{\ell-1} p(\mathcal{P}_{\xi-1}, \mathcal{P}_\xi) \cdot p(\mathcal{P}_{\ell-1}, y) \right] \\ &= p^\ell(x, y), \quad x \in C_i, y \in C_j, \end{aligned}$$

and the first part of the lemma is proved.

The probability $\Pr[x \xrightarrow{\mathcal{P}^\ell} C_j]$, which was defined in Eq. (3.2), combines all the probabilities in Eq. (3.1) with $y \in C_j$. Different paths are considered independent events, thus

$$\Pr[x \xrightarrow{\mathcal{P}^\ell} C_j] = \sum_{y \in C_j} \Pr[x \xrightarrow{\mathcal{P}^\ell} y] = \sum_{y \in C_j} p^\ell(x, y), \quad x \in C_i,$$

and the second part of the lemma is proved. The probability $\Pr[C_i \xrightarrow{\mathcal{P}^\ell} C_j]$, which was defined in Eq. (3.3), combines all the probabilities in Eq. (3.2) with $x \in C_i$. Since x is part of the condition in these probabilities, we get

$$\begin{aligned} \Pr[C_i \xrightarrow{\mathcal{P}^\ell} C_j] &= \sum_{x \in C_i} \Pr[s(\mathcal{P}) = x | s(\mathcal{P}) \in C_i] \cdot \Pr[x \xrightarrow{\mathcal{P}^\ell} C_j] \\ &= \sum_{x \in C_i} \sum_{y \in C_j} \Pr[s(\mathcal{P}) = x | s(\mathcal{P}) \in C_i] \cdot p^\ell(x, y). \end{aligned}$$

Using Eq. (2.3) we get

$$\Pr[C_i \xrightarrow{\mathcal{P}^\ell} C_j] = \sum_{x \in C_i} \sum_{y \in C_j} \frac{q(x)}{\text{vol}(C_i)} p^\ell(x, y),$$

and the final part of the lemma is proved. \square

References

- [1] G. David, A. Averbuch, Hierarchical data organization, clustering and denoising via localized diffusion folders, Appl. Comput. Harmon. Anal. (2012), doi:10.1016/j.acha.2011.09.002, in press.
- [2] R. Coifman, S. Lafon, Diffusion maps, Appl. Comput. Harmon. Anal. 21 (1) (2006) 5–30.

- [3] R. Coifman, S. Lafon, A. Lee, M. Maggioni, B. Nadler, F. Warner, S. Zucker, Geometric diffusions as a tool for harmonic analysis and structure definition of data: Diffusion maps, *Proc. Natl. Acad. Sci. USA* 102 (21) (2005) 7426–7431.
- [4] T. Zhang, R. Ramakrishnan, M. Livny, BIRCH: An efficient data clustering method for very large databases, in: *SIGMOD'96: Proceedings of the 1996 ACM SIGMOD International Conference on Management of Data*, ACM, New York, 1996, pp. 103–114.
- [5] L. Kaufman, P. Rousseeuw, *Finding Groups in Data – An Introduction to Cluster Analysis*, John Wiley & Sons, New York, 1990.
- [6] S. Lafon, A. Lee, Diffusion maps and coarse-graining: A unified framework for dimensionality reduction, graph partitioning, and data set parameterization, *IEEE Trans. Pattern Anal. Mach. Intell.* (2006) 1393–1403.
- [7] M. Salhov, G. Wolf, A. Averbuch, Patch-to-tensor embedding, *Appl. Comput. Harmon. Anal.* (2012), doi:10.1016/j.acha.2011.11.003, in press.
- [8] G. Wolf, A. Averbuch, Linear-projection diffusion on smooth Euclidean submanifolds, *Appl. Comput. Harmon. Anal.*, submitted for publication.
- [9] A. Singer, H. Wu, Orientability and diffusion maps, *Appl. Comput. Harmon. Anal.* 31 (1) (2011) 44–58.
- [10] A. Singer, H. Wu, Vector diffusion maps and the connection Laplacian, preprint, arXiv:1102.0075.
- [11] F. Chung, *Spectral Graph Theory*, CBMS, vol. 92, American Mathematical Society, 1997.